1 Typing Rules

$\boxed{\Theta; \Delta; \Gamma \vdash e : t}$ Typing rules for expressions

$$\frac{}{\Theta; \Delta; \cdot, x : t \vdash x : t} \quad \textsc{Ty\_Var\_Lin}$$

$$\frac{x : t \in \Delta}{\Theta; \Delta; \cdot \vdash x : t} \quad \textsc{Ty\_Var}$$

$$\frac{\begin{array}{l}\Theta; \Delta; \Gamma \vdash e : t \\ \Theta; \Delta; \Gamma', x : t \vdash e' : t'\end{array}}{\Theta; \Delta; \Gamma, \Gamma' \vdash \mathbf{let}\, x = e\, \mathbf{in}\, e' : t'} \quad \textsc{Ty\_Let}$$

$$\frac{}{\Theta; \Delta; \cdot \vdash () : \mathbf{unit}} \quad \textsc{Ty\_Unit\_Intro}$$

$$\frac{\begin{array}{l}\Theta; \Delta; \Gamma \vdash e : \mathbf{unit} \\ \Theta; \Delta; \Gamma' \vdash e' : t\end{array}}{\Theta; \Delta; \Gamma, \Gamma' \vdash \mathbf{let}\, () = e\, \mathbf{in}\, e' : t} \quad \textsc{Ty\_Unit\_Elim}$$

$$\frac{}{\Theta; \Delta; \cdot \vdash \mathbf{true} : \mathbf{bool}} \quad \textsc{Ty\_Bool\_True}$$

$$\frac{}{\Theta; \Delta; \cdot \vdash \mathbf{false} : \mathbf{bool}} \quad \textsc{Ty\_Bool\_False}$$

$$\frac{\begin{array}{l}\Theta; \Delta; \Gamma \vdash e : !\mathbf{bool} \\ \Theta; \Delta; \Gamma' \vdash e_1 : t' \\ \Theta; \Delta; \Gamma' \vdash e_2 : t'\end{array}}{\Theta; \Delta; \Gamma, \Gamma' \vdash \mathbf{if}\, e\, \mathbf{then}\, e_1\, \mathbf{else}\, e_2 : t} \quad \textsc{Ty\_Bool\_Elim}$$

$$\frac{}{\Theta; \Delta; \cdot \vdash k : \mathbf{int}} \quad \textsc{Ty\_Int\_Intro}$$

$$\frac{}{\Theta; \Delta; \cdot \vdash el : \mathbf{elt}} \quad \textsc{Ty\_Elt\_Intro}$$

$$\frac{\begin{array}{l}\Theta; \Delta; \cdot \vdash v : t \\ v \neq l\end{array}}{\Theta; \Delta; \cdot \vdash \mathbf{Many}\, v : !t} \quad \textsc{Ty\_Bang\_Intro}$$

$$\frac{\begin{array}{l}\Theta; \Delta; \Gamma \vdash e : !t \\ \Theta; \Delta, x : t; \Gamma' \vdash e' : t'\end{array}}{\Theta; \Delta; \Gamma, \Gamma' \vdash \mathbf{let}\, \mathbf{Many}\, x = e\, \mathbf{in}\, e' : t'} \quad \textsc{Ty\_Bang\_Elim}$$

$$\frac{\begin{array}{l}\Theta; \Delta; \Gamma \vdash e : t \\ \Theta; \Delta; \Gamma' \vdash e' : t'\end{array}}{\Theta; \Delta; \Gamma, \Gamma' \vdash (e, e') : t \otimes t'} \quad \textsc{Ty\_Pair\_Intro}$$

$$\frac{\begin{array}{l}\Theta; \Delta; \Gamma \vdash e_{12} : t_1 \otimes t_2 \\ \Theta; \Delta; \Gamma', a : t_1, b : t_2 \vdash e : t\end{array}}{\Theta; \Delta; \Gamma, \Gamma' \vdash \mathbf{let}\, (a, b) = e_{12}\, \mathbf{in}\, e : t} \quad \textsc{Ty\_Pair\_Elim}$$

$$\frac{\begin{array}{l}\Theta \vdash t'\ \mathsf{Type} \\ \Theta; \Delta; \Gamma, x : t' \vdash e : t\end{array}}{\Theta; \Delta; \Gamma \vdash \mathbf{fun}\, x : t' \rightarrow e : t' \multimap t} \quad \textsc{Ty\_Lambda}$$

$$\frac{\begin{array}{l}\Theta; \Delta; \Gamma \vdash e : t' \multimap t \\ \Theta; \Delta; \Gamma' \vdash e' : t'\end{array}}{\Theta; \Delta; \Gamma, \Gamma' \vdash e\, e' : t} \quad \textsc{Ty\_App}$$

$$\frac{\Theta, fc; \Delta; \Gamma \vdash e : t}{\Theta; \Delta; \Gamma \vdash \mathbf{fun}\, fc \to e : \forall fc.t} \quad \textsc{Ty\_Gen}$$

$$\frac{\begin{array}{l}\Theta \vdash f\ \mathsf{Cap} \\ \Theta; \Delta; \Gamma \vdash e : \forall fc.t\end{array}}{\Theta; \Delta; \Gamma \vdash e[f] : t[f/fc]} \quad \textsc{Ty\_Spc}$$

$$\frac{\Theta; \Delta, g : t \multimap t'; \cdot, x : t \vdash e : t'}{\Theta; \Delta; \cdot \vdash \mathbf{fix}\,(g, x : t, e : t') : !(t \multimap t')} \quad \textsc{Ty\_Fix}$$

2 Operational Semantics

$\boxed{\langle \sigma, e \rangle \to \mathit{StepsTo}}$ operational semantics

$$\frac{}{\langle \sigma, \textbf{let}\ () = ()\ \textbf{in}\ e \rangle \to \langle \sigma, e \rangle}\ \textsc{Op\_Let\_Unit}$$

$$\frac{}{\langle \sigma, \textbf{let}\ x = v\ \textbf{in}\ e \rangle \to \langle \sigma, e[x/v] \rangle}\ \textsc{Op\_Let\_Var}$$

$$\frac{}{\langle \sigma, \textbf{if}\ (\textbf{Many}\ \textbf{true})\ \textbf{then}\ e_1\ \textbf{else}\ e_2 \rangle \to \langle \sigma, e_1 \rangle}\ \textsc{Op\_If\_True}$$

$$\frac{}{\langle \sigma, \textbf{if}\ (\textbf{Many}\ \textbf{false})\ \textbf{then}\ e_1\ \textbf{else}\ e_2 \rangle \to \langle \sigma, e_2 \rangle}\ \textsc{Op\_If\_False}$$

$$\frac{}{\langle \sigma, \textbf{let}\ \textbf{Many}\ x = \textbf{Many}\ v\ \textbf{in}\ e \rangle \to \langle \sigma, e[x/v] \rangle}\ \textsc{Op\_Let\_Many}$$

$$\frac{}{\langle \sigma, \textbf{let}\ (a, b) = (v_1, v_2)\ \textbf{in}\ e \rangle \to \langle \sigma, e[a/v_1][b/v_2] \rangle}\ \textsc{Op\_Let\_Pair}$$

$$\frac{e_1 = e[g/\textbf{fun}\ x : t \to \textbf{let}\ \textbf{Many}\ g = \textbf{fix}\ (g, x : t, e : t')\ \textbf{in}\ g\ x]}{\langle \sigma, \textbf{let}\ \textbf{Many}\ g = \textbf{fix}\ (g, x : t, e : t')\ \textbf{in}\ e' \rangle \to \langle \sigma, e'[g/\textbf{fun}\ x : t \to e_1] \rangle}\ \textsc{Op\_Let\_Fix}$$

$$\frac{}{\langle \sigma, (\textbf{fun}\ fc \to v)[f] \rangle \to \langle \sigma, v[fc/f] \rangle}\ \textsc{Op\_Frac\_Cap}$$

$$\frac{}{\langle \sigma, (\textbf{fun}\ x : t \to e)\ v \rangle \to \langle \sigma, e[x/v] \rangle}\ \textsc{Op\_App}$$

$$\frac{\langle \sigma, e \rangle \to \langle \sigma', e' \rangle}{\langle \sigma, C[e] \rangle \to \langle \sigma, C[e'] \rangle}\ \textsc{Op\_Context}$$

$$\frac{\langle \sigma, e \rangle \to \textbf{err}}{\langle \sigma, C[e] \rangle \to \textbf{err}}\ \textsc{Op\_Context\_Err}$$

$$\frac{0 \le k_1, k_2}{\langle \sigma, \textbf{matrix}\ k_1\ k_2 \rangle \to \langle \sigma \uplus \{l \mapsto_1 M_{k_1,k_2}\}, l \rangle}\ \textsc{Op\_Matrix}$$

$$\frac{}{\langle \sigma \uplus \{l \mapsto_1 m_{k_1,k_2}\}, \textbf{free}\ l \rangle \to \langle \sigma, () \rangle}\ \textsc{Op\_Free}$$

$$\frac{}{\langle \sigma \uplus \{l \mapsto_f m_{k_1,k_2}\}, \textbf{share}\ l \rangle \to \langle \sigma \uplus \{l \mapsto_{\frac{1}{2}\cdot f} m_{k_1,k_2}\} \uplus \{l \mapsto_{\frac{1}{2}\cdot f} m_{k_1,k_2}\}, (l, l) \rangle}\ \textsc{Op\_Share}$$

$$\frac{f \le 1}{\langle \sigma \uplus \{l \mapsto_{\frac{1}{2}\cdot f} m_{k_1,k_2}\} \uplus \{l \mapsto_{\frac{1}{2}\cdot f} m_{k_1,k_2}\}, \textbf{unshare}\ l\ l \rangle \to \langle \sigma \uplus \{l \mapsto_f m_{k_1,k_2}\}, l \rangle}\ \textsc{Op\_Unshare\_Eq}$$

$$\frac{l \ne l'}{\langle \sigma \uplus \{l \mapsto_{\frac{1}{2}\cdot f} m_{k_1,k_2}\} \uplus \{l' \mapsto_{\frac{1}{2}\cdot f} m_{k_1,k_2}\}, \textbf{unshare}\ l\ l' \rangle \to \textbf{err}}\ \textsc{Op\_Unshare\_Neq}$$

$$\frac{\sigma' = \sigma \uplus \{l_1 \mapsto_{fc_1} m1_{k_1,k_2}\} \uplus \{l_2 \mapsto_{fc_2} m2_{k_2,k_3}\}}{\langle \sigma' \uplus \{l_3 \mapsto_1 m1_{k_1,k_3}\}, \textbf{gemm}\ l_1\ l_2\ l_3 \rangle \to \langle \sigma' \uplus \{l_3 \mapsto_1 (m_1\ m_2 + m_3)_{k_1,k_3}\}, ((l_1, l_2), l_3) \rangle}\ \textsc{Op\_Gemm\_Match}$$

$$\frac{\begin{array}{l} k_2 \ne k_2' \\ \sigma' = \sigma \uplus \{l_1 \mapsto_{fc_1} m1_{k_1,k_2}\} \uplus \{l_2 \mapsto_{fc_2} m2_{k_2',k_3}\} \end{array}}{\langle \sigma' \uplus \{l_3 \mapsto_1 m1_{k_1,k_3}\}, \textbf{gemm}\ l_1\ l_2\ l_3 \rangle \to \textbf{err}}\ \textsc{Op\_Gemm\_Mismatch}$$

3 Interpretation

$$\mathcal{V}_k[\![\textbf{unit}]\!] = \{(\emptyset, *)\}$$

$$\mathcal{V}_k[\![\textbf{bool}]\!] = \{(\emptyset, true), (\emptyset, false)\}$$

$$\mathcal{V}_k[\![\textbf{int}]\!] = \{(\emptyset, n) \mid 2^{-63} \leq n \leq 2^{63} - 1\}$$

$$\mathcal{V}_k[\![\textbf{elt}]\!] = \{(\emptyset, f) \mid f \text{ a IEEE Float64 }\}$$

$$\mathcal{V}_k[\![f\,\textbf{mat}]\!] = \{(\{l \mapsto_{2^{-f}} \_\}, l)\}$$

$$\begin{aligned}\mathcal{V}_k[\![!(t' \multimap t'')]\!] = &\{(\emptyset, \textbf{Many}\, v) \mid (\emptyset, v) \in \mathcal{V}_k[\![t' \multimap t'']\!]\}\\ &\cup \{(\emptyset, \textbf{fix}(g, x : t, e : t')) \mid \forall\, j < k, (\sigma, v) \in \mathcal{V}_j[\![t]\!].\\ &\qquad (\sigma, \textbf{let}\,\textbf{Many}\, g = \textbf{fix}\,(g, x : t, e : t')\,\textbf{in}\, g\, v) \in \mathcal{C}_j[\![t']\!]\}\end{aligned}$$

$$\mathcal{V}_k[\![!t]\!] = \{(\emptyset, \textbf{Many}\, v) \mid \neg(\exists t', t''.\ t = t' \multimap t'') \wedge (\emptyset, v) \in \mathcal{V}_k[\![t]\!]\}$$

$$\mathcal{V}_k[\![\forall fc.\ t]\!] = \{(\sigma, \textbf{fun}\, fc \rightarrow\ v) \mid \forall f.\ (\sigma, (\textbf{fun}\, fc \rightarrow\ v)\,[f]) \in \mathcal{V}_k[\![t[fc/f]]\!]\}$$

$$\mathcal{V}_k[\![t_1 \otimes t_2]\!] = \{(\sigma_1 \star \sigma_2, (v_1, v_2)) \mid (\sigma_1, v_1) \in \mathcal{V}_k[\![t_1]\!] \wedge (\sigma_2, v_2) \in \mathcal{V}_k[\![t_2]\!]\}$$

$$\begin{aligned}\mathcal{V}_k[\![t \multimap t']\!] = &\{(\sigma, \textbf{fun}\, x : t \rightarrow e) \mid \forall j < k, (\sigma', v') \in \mathcal{V}_j[\![t']\!].\ \sigma \star \sigma' \text{ defined } \Rightarrow\\ &\qquad (\sigma \star \sigma', (\textbf{fun}\, x : t \rightarrow e)\, v') \in \mathcal{C}_j[\![t']\!]\}\end{aligned}$$

$$\begin{aligned}\mathcal{C}_k[\![t]\!] = &\{(\sigma_s, e) \mid \forall\, j \leq k, \sigma_r.\ \sigma_s \star \sigma_r \text{ defined } \Rightarrow \langle \sigma_s \star \sigma_r, e\rangle \rightarrow^j \textbf{err}\ \vee \exists \sigma_f, e'.\\ &\qquad \langle \sigma_s \star \sigma_r, e\rangle \rightarrow^j \langle \sigma_f \star \sigma_r, e'\rangle \wedge (e' \text{ is a value } \Rightarrow (\sigma_f \star \sigma_r, e') \in \mathcal{V}_{k-j}[\![t]\!])\}\end{aligned}$$

$$\mathcal{I}_k[\![\cdot]\!]\theta = \{[]\}$$

$$\mathcal{I}_k[\![\Delta, x : t]\!]\theta = \{\delta[x \mapsto v_x] \mid \delta \in \mathcal{I}_k[\![\Delta]\!]\theta \wedge (\emptyset, v_x) \in \mathcal{V}_k[\![\theta(t)]\!]\}$$

$$\mathcal{L}_k[\![\cdot]\!]\theta = \{(\emptyset, [])\}$$

$$\mathcal{L}_k[\![\Gamma, x : t]\!]\theta = \{(\sigma \star \sigma_x, \gamma[x \mapsto v_x]) \mid (\sigma, \gamma) \in \mathcal{L}_k[\![\Gamma]\!]\theta \wedge (\sigma_x, v_x) \in \mathcal{V}_k[\![\theta(t)]\!]\}$$

$$\begin{aligned}[\![\Theta; \Delta; \Gamma \vdash e : t]\!] = &\forall \theta, k, \delta, \gamma, \sigma.\ \mathrm{dom}(\Theta) = \mathrm{dom}(\theta) \wedge (\sigma, \gamma) \in \mathcal{L}_k[\![\Gamma]\!]\theta \wedge \delta \in \mathcal{I}_k[\![\Delta]\!]\theta \Rightarrow\\ &\qquad (\sigma, \gamma(\delta(e))) \in \mathcal{C}_k[\![\theta(t)]\!]\end{aligned}$$

4 Soundness Proof

$$\forall \Theta, \Delta, \Gamma, e, t.\ \Theta; \Delta; \Gamma \vdash e : t \Rightarrow [\![\Theta; \Delta; \Gamma \vdash e : t]\!]$$

Proof sketch: Induction over the typing judgements.

Assume: 1. Arbitrary $\Theta, \Delta, \Gamma, e, t$ such that $\Theta; \Delta; \Gamma \vdash e : t$.
2. Arbitrary $\theta, k, \delta, \gamma, \sigma$ such that:
 a. $\mathrm{dom}(\Theta) = \mathrm{dom}(\theta)$
 b. $(\sigma, \gamma) \in \mathcal{L}_k[\![\Gamma]\!]\theta$
 c. $\delta \in \mathcal{I}_k[\![\Delta]\!]\theta$.
3. W.l.o.g., all variables are distinct/dom(Δ) and dom(Γ) are disjoint.
4. And so that over expressions $\gamma \circ \delta = \delta \circ \gamma$.
5. By construction, $\mathrm{dom}(\Delta) = \mathrm{dom}(\delta)$ and $\mathrm{dom}(\Gamma) = \mathrm{dom}(\gamma)$.
6. ??? $\mathcal{V}_k[\![\theta(t)]\!] \subseteq \mathcal{C}_k[\![\theta(t)]\!]$.
7. ??? "Stronger heap"/frame rule: $\langle \sigma, e \rangle \rightarrow^* = \langle \sigma \star \sigma_r, e \rangle \rightarrow^*$.
8. ??? $\delta(\gamma(v))$ is a value.
9. ??? $j \leq k \Rightarrow {}_{-\,k}[\![\cdot]\!] \subseteq {}_{-\,j}[\![\cdot]\!]$
10. ??? $(\sigma', e') \in \mathcal{C}_{k-1}[\![\cdot]\!] \wedge \langle \sigma, e \rangle \rightarrow \langle \sigma', e' \rangle \Rightarrow (\sigma, e) \in \mathcal{C}_k[\![\cdot]\!]$

Prove: $(\sigma, \gamma(\delta(e))) \in \mathcal{C}_k[\![\theta(t')]\!]$.
Assume: Arbitrary $j \leq k$ and σ_r.
Suffices: Show whole expression either reduces to **err** or takes j steps.

⟨1⟩1. Case: Ty_Let.
Prove: $(\sigma, \gamma(\delta(\mathbf{let}\, x\ =\ e\, \mathbf{in}\, e'))) \in \mathcal{C}_k[\![\theta(t')]\!]$.
Suffices: $(\sigma, \mathbf{let}\, x\ =\ \gamma(\delta(e))\, \mathbf{in}\, \gamma(\delta(e'))) \in \mathcal{C}_k[\![\theta(t')]\!]$.

⟨2⟩1. By induction,
1. $[\![\Theta; \Delta; \Gamma \vdash e : t]\!]$
2. $[\![\Theta; \Delta; \Gamma', x : t \vdash e' : t']\!]$.

⟨2⟩2. By 2b and induction on Γ', we know there exist $\sigma_{e'}$, $(\sigma_e, \gamma_e) \in \mathcal{L}_k[\![\Gamma]\!]$, such that $\sigma = \sigma_e \star \sigma_{e'}$.

⟨2⟩3. So, using them, θ, k, δ, and 3 we have $(\sigma_e, \gamma_e(e)) \in \mathcal{C}_k[\![\theta(t)]\!]$.

⟨2⟩4. By 3, $(\sigma_e, \gamma(\delta(e))) \in \mathcal{C}_k[\![\theta(t)]\!]$.

⟨2⟩5. By definition of $\mathcal{C}_k[\![\cdot]\!]$ and ⟨2⟩2, we instantiate with j and $\sigma_r = \sigma_{e'}$ to conclude that $\langle \sigma, \gamma(\delta(e)) \rangle$ either reduces to **err** or another heap and expression.

⟨2⟩6. Case: **err**
??? By Op_Context_Err and 7 with σ_r, the whole expression reduces to **err** in $j \leq k$ steps. Since $j \leq k$ and σ_r are arbitrary, $(\sigma, \gamma(\delta(\mathbf{let}\, x\ =\ e\, \mathbf{in}\, e'))) \in \mathcal{C}_k[\![\theta(t')]\!]$.

⟨2⟩7. Case: j steps to another heap and expression.
By Op_Context, the whole expression does the same.

⟨2⟩8. If it is not a value, we are done. ??? If it is $(\sigma_{ef}, v) \in \mathcal{V}_{k-j}[\![\theta(t)]\!]$ by 8.
Suffices: $(\sigma_{ef} \star \sigma_{e'}, \mathbf{let}\, x\ =\ v\, \mathbf{in}\, \gamma(\delta(e'))) \in \mathcal{C}_{k-j}[\![\theta(t')]\!]$.
Suffices: ??? $(\sigma_{ef} \star \sigma_{e'}, \gamma(\delta(e'))[x/v]) \in \mathcal{C}_{k-j-1}[\![\theta(t')]\!]$ by 10.

⟨2⟩9. Define: $\gamma_{e'}(y) = v$ if $y = x$ and $\gamma(y)$ if $y \in \mathrm{dom}(\Gamma')$.

??? Thus, by 9, $(\sigma_{e'}, \gamma_{e'}) \in \mathcal{L}_k[\![\Gamma', x : t]\!]\theta \subseteq \mathcal{L}_{k-j-1}[\![\Gamma', x : t]\!]\theta$.

$\langle 2\rangle$10. Instantiate 2 of step $\langle 2\rangle$1 with $\theta, k-j-1, \delta, \gamma_{e'}, \sigma_{e'}$ to conclude $(\sigma_{e'}, \gamma_{e'}(\delta(e'))) \in \mathcal{C}_{k-j-1}[\![\theta(t')]\!]$.

$\langle 2\rangle$11. By 3, we have $\gamma(\delta(e'))[x/v] = \gamma_{e'}(\delta(e'))$ and by 7 we conclude $(\sigma_{ef} \star \sigma_{e'}, \gamma(\delta(e'))[x/v]) \in \mathcal{C}_{k-j-1}[\![\theta(t')]\!]$

$\langle 1\rangle$2. CASE: TY_PAIR_ELIM.
PROVE: $(\sigma, \gamma(\delta(\mathbf{let}\,(a, b) \;=\; e\,\mathbf{in}\,e'))) \in \mathcal{C}_k[\![\theta(t')]\!]$.
PROOF: Similar to TY_LET but with OP_LET_PAIR

$\langle 2\rangle$1. When $(\sigma_{ef}, v) \in \mathcal{V}_{k-j}[\![\theta(t_1) \otimes \theta(t_2)]\!]$, we have $v = (v_1, v_2)$.

$\langle 2\rangle$2. SUFFICES: ??? $(\sigma_{e'}, \gamma(\delta(e'))) \in \mathcal{C}_{k-j-1}[\![\theta(t')]\!]$ by 10.

$\langle 2\rangle$3. DEFINE: $\gamma_{e'}$ to be the restriction of γ to $\mathrm{dom}(\Gamma')$.
??? Thus, by 9, $(\sigma_{e'}, \gamma_{e'}[a \mapsto v_1, b \mapsto v_2]) \in \mathcal{L}_k[\![\Gamma', a : t_1, b : t_2]\!]\theta$
$\subseteq \mathcal{L}_{k-j-1}[\![\Gamma', a : t_1, b : t_2]\!]\theta$.

$\langle 2\rangle$4. Instantiate $[\![\Theta; \Delta; \Gamma', a : t_1, b : t_2 \vdash e' : t']\!]$ with $\theta, k-j-1, \delta, \gamma_{e'}[a \mapsto v_1, b \mapsto v_2], \sigma_{e'}$.

$\langle 2\rangle$5. ??? By 3 $(\sigma_{e'}, \gamma(\delta(e'))) \in \mathcal{C}_{k-j-1}[\![\theta(t')]\!]$.

$\langle 1\rangle$3. CASE: TY_BANG_ELIM.
PROVE: $(\sigma, \gamma(\delta(\mathbf{let\,Many}\,x \;=\; e\,\mathbf{in}\,e'))) \in \mathcal{C}_k[\![\theta(t)]\!]$.
PROOF SKETCH: Similar to TY_LET, but with the following key differences.

$\langle 2\rangle$1. When $(\sigma_{ef}, v) \in \mathcal{V}_{k-j}[\![\theta(!t)]\!]$, since $\mathcal{V}_{k-j}[\![\theta(!t)]\!] = \mathcal{V}_{k-j}[\![!\theta(t)]\!]$, we have $\sigma_{ef} = \emptyset$ and $v = \mathbf{Many}\,v'$ for some $(\emptyset, v') \in \mathcal{V}_{k-j}[\![\theta(t)]\!]$.

$\langle 2\rangle$2. SUFFICES: $(\sigma_{e'}, \mathbf{let\,Many}\,x \;=\; \mathbf{Many}\,v'\,\mathbf{in}\,\gamma(\delta(e'))) \in \mathcal{C}_{k-j}[\![\theta(t)]\!]$.

$\langle 2\rangle$3. SUFFICES: $(\sigma_{e'}, \gamma(\delta(e'))[x/v]) \in \mathcal{C}_{k-j-1}[\![\theta(t)]\!]$.

$\langle 2\rangle$4. DEFINE: $\gamma_{e'}$ as the restriction of γ to $\mathrm{dom}(\Gamma')$.

$\langle 2\rangle$5. Instantiate $[\![\Theta; \Delta, x : t, \Gamma' \vdash e' : t']\!]$ with $\theta, k-j-1, \delta_{e'} = \delta[x \mapsto v'], \gamma_{e'}, \sigma_{e'}$ to conclude $(\sigma_{e'}, \gamma_{e'}(\delta_{e'}(e'))) \in \mathcal{C}_{k-j-1}[\![\theta(t)]\!]$.

$\langle 2\rangle$6. ??? By 3, $(\sigma_{e'}, \gamma(\delta(e'))[x/v]) \in \mathcal{C}_{k-j-1}[\![\theta(t)]\!]$.

$\langle 1\rangle$4. CASE: TY_UNIT_ELIM.
PROVE: $(\sigma, \gamma(\delta(\mathbf{let}\,() \;=\; e\,\mathbf{in}\,e'))) \in \mathcal{C}_k[\![\theta(t)]\!]$.
PROOF: Similar to TY_LET but with OP_LET_UNIT.

$\langle 2\rangle$1. When $(\sigma_{ef}, v) \in \mathcal{V}_{k-j}[\![\mathbf{unit}]\!]$, we have $\sigma_{ef} = \emptyset$ and $v = ()$.

$\langle 2\rangle$2. SUFFICES: ??? $(\sigma_{e'}, \gamma(\delta(e'))) \in \mathcal{C}_{k-j-1}[\![\theta(t')]\!]$ by 10.

$\langle 2\rangle$3. DEFINE: $\gamma_{e'}$ to be the restriction of γ to $\mathrm{dom}(\Gamma')$.
??? Thus, by 9, $(\sigma_{e'}, \gamma_{e'}) \in \mathcal{L}_k[\![\Gamma']\!]\theta \subseteq \mathcal{L}_{k-j-1}[\![\Gamma']\!]\theta$.

$\langle 2\rangle$4. Instantiate $[\![\Theta; \Delta; \Gamma' \vdash e' : t']\!]$ with $\theta, k-j-1, \delta, \gamma_{e'}, \sigma_{e'}$.

$\langle 2\rangle$5. ??? By 3 $(\sigma_{e'}, \gamma(\delta(e'))) \in \mathcal{C}_{k-j-1}[\![\theta(t')]\!]$.

⟨1⟩5. CASE: TY_BOOL_ELIM.
PROVE: $(\sigma, \gamma(\delta(\textbf{if}\, e\, \textbf{then}\, e_1\, \textbf{else}\, e_2))) \in \mathcal{C}_k[\![\theta(t)]\!]$.
PROOF: Similar to TY_UNIT_ELIM but with OP_IF_{TRUE,FALSE} and $\sigma_{ef} = \emptyset$ and $v = \textbf{Many true}$ or $v = \textbf{Many false}$.

⟨1⟩6. CASE: TY_BANG_INTRO.
PROVE: $(\sigma, \gamma(\delta(\textbf{Many}\, e))) \in \mathcal{C}_k[\![\theta(!t)]\!]$.
SUFFICES: $(\sigma, \textbf{Many}\, \gamma(\delta(e))) \in \mathcal{C}_k[\![!\theta(t)]\!]$.

⟨2⟩1. By assumption of TY_BANG_INTRO, $e = v$ for some value $v \neq l$, $\Gamma = \emptyset$ and so $[\![\Theta; \Delta; \cdot \vdash v : t]\!]$ by induction.

⟨2⟩2. SUFFICES: $(\emptyset, \textbf{Many}\, \delta(v)) \in \mathcal{C}_k[\![!\theta(t)]\!]$ by 3 and 2b.

⟨2⟩3. Instantiate $[\![\Theta; \Delta; \cdot \vdash v : t]\!]$ with $\theta, k, \delta, \gamma = [], \sigma = \emptyset$ to obtain $(\emptyset, \delta(v)) \in \mathcal{C}_k[\![\theta(t)]\!]$.

⟨2⟩4. Instantiate $(\emptyset, \delta(v)) \in \mathcal{C}_k[\![\theta(t)]\!]$ with $j = 0$, and $\sigma_r = \emptyset$, to conclude $(\emptyset, v) \in \mathcal{V}_k[\![\theta(t)]\!]$.

⟨2⟩5. ??? By definition of $\mathcal{V}_k[\![!\theta(t)]\!]$, 8 and 6 we have $(\emptyset, \textbf{Many}\, \delta(v)) \in \mathcal{C}_k[\![!\theta(t)]\!]$.

⟨1⟩7. CASE: TY_PAIR_INTRO.
PROVE: $(\sigma, \gamma(\delta(\,(e, e')\,))) \in \mathcal{C}_k[\![\theta(t \otimes t')]\!]$.

⟨2⟩1. By induction,
1. $[\![\Theta; \Delta; \Gamma \vdash e : t]\!]$
2. $[\![\Theta; \Delta; \Gamma' \vdash e' : t']\!]$.

⟨2⟩2. First component reduces to error in j_1 steps, so the whole thing does too.

⟨2⟩3. First component reduces to value in j_1 steps, so ....

⟨2⟩4. Second component reduces to error in j_2 steps, so the whole thing does too.

⟨2⟩5. Second component also reduces to value in j_2 steps, so ....

⟨2⟩6. Should be possible by IH instantiating with $k - j_2, k - j_1$ (swap, the subtract remaining) respectively.

⟨1⟩8. CASE: TY_LAMBDA.
PROVE: $(\sigma, \gamma(\delta(\textbf{fun}\, x : t' \rightarrow e))) \in \mathcal{C}_k[\![\theta(t' \multimap t)]\!]$.

⟨1⟩9. CASE: TY_APP.
PROVE: $(\sigma, \gamma(\delta(\, e\, e'\,))) \in \mathcal{C}_k[\![\theta(t)]\!]$.

⟨1⟩10. CASE: TY_GEN.
PROVE: $(\sigma, \gamma(\delta(\textbf{fun}\, fc \rightarrow e))) \in \mathcal{C}_k[\![\theta(\forall\, fc.\, t)]\!]$.

⟨1⟩11. CASE: TY_SPC.
PROVE: $(\sigma, \gamma(\delta(e\,[f]))) \in \mathcal{C}_k[\![\theta(t\,[fc/f])]\!]$.

⟨1⟩12. CASE: TY_FIX.
PROVE: $(\sigma, \gamma(\delta(\mathbf{fix}(g, x : t, e : t')))) \in \mathcal{C}_k[\![\theta(!(t \multimap t'))]\!]$. This means $\sigma = \emptyset$.
SUFFICES: ??? to show $\ldots \in \mathcal{V}_k[\![!(\theta(t) \multimap \theta(t'))]\!]$, by 6.

⟨2⟩1. ASSUME: Arbitrary $j < k$ and $(\sigma, v) \in \mathcal{V}_j[\![\theta(t)]\!]$.

⟨2⟩2. SUFFICES: $(\sigma, \mathbf{let\,Many}\; g = \mathbf{fix}\,(g, x : t, e : t')\,\mathbf{in}\, g\, v) \in \mathcal{C}_j[\![\theta(t')]\!]$.

⟨2⟩3. LET: $e_1 = e[g/\mathbf{fun}\, x : t \rightarrow \mathbf{let\,Many}\; g = \mathbf{fix}\,(g, x : t, e : t')\,\mathbf{in}\, g\, x]$.

⟨2⟩4. SUFFICES: ??? by 10, $(\sigma, (\mathbf{fun}\, x : t \rightarrow e_1)\, v) \in \mathcal{C}_{j-1}[\![\theta(t')]\!]$.

⟨2⟩5. SUFFICES: ??? by 10, $(\sigma, e_1[x/v]) \in \mathcal{C}_{j-2}[\![\theta(t')]\!]$.

⟨2⟩6. By induction, we have $[\![\Theta; \Delta, g : t \multimap t'; x : t \vdash e : t']\!]$.

⟨2⟩7. Instantiate this with $\theta, j-2, \delta[g \mapsto \mathbf{fun}\, x : t \rightarrow e_1], \gamma = [x \mapsto v], \sigma = \emptyset$.
PROVE: $(\emptyset, \mathbf{fun}\, x : t \rightarrow e_1) \in \mathcal{V}_{j-2}[\![\theta(t) \multimap \theta(t')]\!]$.

⟨3⟩1. SUFFICES: ??? by 10, $(\sigma', e_1[x/v']) \in \mathcal{C}_{j-2}[\![\theta(t')]\!]$ for arbitrary $(\sigma', v') \in \mathcal{V}_{j-2}[\![\theta(t)]\!]$.

⟨3⟩2. We can again use the induction hypothesis $[\![\Theta; \Delta, g : t \multimap t'; x : t \vdash e : t']\!]$.

⟨3⟩3. But since it's true for $\mathcal{C}_0[\![\cdot]\!]$ (base case), it's true by induction ???

⟨2⟩8. Lastly, we show $\delta(\gamma(e)) = e_1[x/v]$, which follows by their definitions,
to conclude $(\sigma, e_1[x/v]) \in \mathcal{C}_{j-2}[\![\theta(t')]\!]$.

⟨1⟩13. CASE: TY_VAR_LIN.
PROVE: $(\sigma, \gamma(\delta(x))) \in \mathcal{C}_k[\![\theta(t)]\!]$.

⟨2⟩1. $\Gamma = \{x : t\}$ by assumption of TY_VAR_LIN.

⟨2⟩2. SUFFICES: $(\sigma, \gamma(x)) \in \mathcal{C}_k[\![\theta(t)]\!]$ by 3.

⟨2⟩3. By 2b, there exist $(\sigma_x, v_x) \in \mathcal{V}_k[\![\theta(t)]\!]$, such that $\sigma = \sigma_x$ and $\gamma = [x \mapsto v_x]$.

⟨2⟩4. ??? Hence, $(\sigma_x, v_x) \in \mathcal{C}_k[\![\theta(t)]\!]$, by 6.

⟨1⟩14. CASE: TY_VAR.
PROVE: $(\sigma, \gamma(\delta(x))) \in \mathcal{C}_k[\![\theta(t)]\!]$.

⟨2⟩1. $x : t \in \Delta$ and $\Gamma = \emptyset$ by assumption of TY_VAR.

⟨2⟩2. SUFFICES: $(\emptyset, \delta(x)) \in \mathcal{C}_k[\![\theta(t)]\!]$ by 3 and 2b.

⟨2⟩3. By 2c, there exists v_x such that $(\emptyset, v_x) \in \mathcal{V}_k[\![\theta(t)]\!]$.

⟨2⟩4. ??? Hence, $(\emptyset, v_x) \in \mathcal{C}_k[\![\theta(t)]\!]$, by 6.

⟨1⟩15. CASE: TY_UNIT_INTRO.
PROVE: $(\sigma, \gamma(\delta(\,()\,))) \in \mathcal{C}_k[\![\theta(\mathbf{unit})]\!]$.

⟨1⟩16. CASE: TY_BOOL_TRUE, TY_BOOL_FALSE, TY_INT_INTRO, TY_ELT_INTRO.
Similar to TY_UNIT_INTRO.

5 Grammar Definition

m	$::=$			matrix expressions
	$\|$	M		matrix variables
	$\|$	$m + m'$		matrix addition
	$\|$	$m\, m'$		matrix multiplication
	$\|$	(m)	S	
f	$::=$			fractional capability
	$\|$	fc		variable
	$\|$	1		whole capability
	$\|$	$\frac{1}{2} \cdot f$		
t	$::=$			linear type
	$\|$	**unit**		unit
	$\|$	**bool**		boolean (true/false)
	$\|$	**int**		63-bit integers
	$\|$	**elt**		array element
	$\|$	f **arr**		arrays
	$\|$	f **mat**		matrices
	$\|$	$!t$		multiple-use type
	$\|$	$\forall\, fc.t$	bind fc in t	frac. cap. generalisation
	$\|$	$t \otimes t'$		pair
	$\|$	$t \multimap t'$		linear function
	$\|$	(t)	S	parentheses
p	$::=$			primitive
	$\|$	**not**		boolean negation
	$\|$	$(+)$		integer addition
	$\|$	$(-)$		integer subtraction
	$\|$	$(*)$		integer multiplication
	$\|$	$(/)$		integer division
	$\|$	$(=)$		integer equality
	$\|$	$(\langle)$		integer less-than
	$\|$	$(+.)$		element addition
	$\|$	$(-.)$		element subtraction
	$\|$	$(*.)$		element multiplication
	$\|$	$(/.)$		element division
	$\|$	$(=.)$		element equality
	$\|$	$(<.)$		element less-than
	$\|$	**set**		array index assignment
	$\|$	**get**		array indexing
	$\|$	**share**		share array
	$\|$	**unshare**		unshare array
	$\|$	**free**		free arrary
	$\|$	**array**		Owl: make array
	$\|$	**copy**		Owl: copy array
	$\|$	**sin**		Owl: map sine over array

	$\mid$	**hypot**		Owl: $x_i := \sqrt{x_i^2 + y_i^2}$
	$\mid$	**asum**		BLAS: $\sum_i \lvert x_i \rvert$
	$\mid$	**axpy**		BLAS: $x := \alpha x + y$
	$\mid$	**dot**		BLAS: $x \cdot y$
	$\mid$	**rotmg**		BLAS: see its docs
	$\mid$	**scal**		BLAS: $x := \alpha x$
	$\mid$	**amax**		BLAS: argmax $i : x_i$
	$\mid$	**setM**		matrix index assignment
	$\mid$	**getM**		matrix indexing
	$\mid$	**shareM**		share matrix
	$\mid$	**unshareM**		unshare matrix
	$\mid$	**freeM**		free matrix
	$\mid$	**matrix**		Owl: make matrix
	$\mid$	**copyM**		Owl: copy matrix
	$\mid$	**copyM_to**		Owl: copy matrix onto another
	$\mid$	**sizeM**		dimension of matrix
	$\mid$	**trnsp**		transpose matrix
	$\mid$	**gemm**		BLAS: $C := \alpha A^{T?} B^{T?} + \beta C$
	$\mid$	**symm**		BLAS: $C := \alpha AB + \beta C$
	$\mid$	**posv**		BLAS: Cholesky decomp. and solve
	$\mid$	**potrs**		BLAS: solve with given Cholesky
v	$::=$			values
	$\mid$	p		primitives
	$\mid$	x		variable
	$\mid$	$()$		unit introduction
	$\mid$	**true**		true
	$\mid$	**false**		false
	$\mid$	k		integer
	$\mid$	l		heap location
	$\mid$	el		array element
	$\mid$	**Many** v		!-introduction
	$\mid$	**fun** $fc \to v$		frac. cap. abstraction
	$\mid$	$v[f]$		frac. cap. specialisation
	$\mid$	(v, v')		pair introduction
	$\mid$	**fun** $x : t \to e$	bind x in e	abstraction
	$\mid$	**fix** $(g, x : t, e : t')$	bind $g \cup x$ in e	fixpoint
	$\mid$	(v)	S	parentheses
e	$::=$			expression
	$\mid$	p		primitives
	$\mid$	x		variable
	$\mid$	**let** $x = e$ **in** e'	bind x in e'	let binding
	$\mid$	$()$		unit introduction
	$\mid$	**let** $() = e$ **in** e'		unit elimination
	$\mid$	**true**		true

$\mid$	$\mathbf{false}$		false
$\mid$	$\mathbf{if}\ e\ \mathbf{then}\ e_1\ \mathbf{else}\ e_2$		if
$\mid$	k		integer
$\mid$	l		heap location
$\mid$	el		array element
$\mid$	$\mathbf{Many}\ e$		!-introduction
$\mid$	$\mathbf{let}\ \mathbf{Many}\ x = e\ \mathbf{in}\ e'$		!-elimination
$\mid$	$\mathbf{fun}\ fc \rightarrow e$		frac. cap. abstraction
$\mid$	$e[f]$		frac. cap. specialisation
$\mid$	(e, e')		pair introduction
$\mid$	$\mathbf{let}\ (a, b) = e\ \mathbf{in}\ e'$	bind $a \cup b$ in e'	pair elimination
$\mid$	$\mathbf{fun}\ x : t \rightarrow e$	bind x in e	abstraction
$\mid$	$e\ e'$		application
$\mid$	$\mathbf{fix}\ (g, x : t, e : t')$	bind $g \cup x$ in e	fixpoint
$\mid$	(e)	S	parentheses

C ::=			evaluation contexts
$\mid$	$\mathbf{let}\ x = [-]\ \mathbf{in}\ e$	bind x in e	let binding
$\mid$	$\mathbf{let}\ () = [-]\ \mathbf{in}\ e$		unit elimination
$\mid$	$\mathbf{if}\ [-]\ \mathbf{then}\ e_1\ \mathbf{else}\ e_2$		if
$\mid$	$\mathbf{Many}\ [-]$		!-introduction
$\mid$	$\mathbf{let}\ \mathbf{Many}\ x = [-]\ \mathbf{in}\ e$		!-elimination
$\mid$	$\mathbf{fun}\ fc \rightarrow [-]$		frac. cap. abstraction
$\mid$	$[-][f]$		frac. cap. specialisation
$\mid$	$([-], e)$		pair introduction
$\mid$	$(v, [-])$		pair introduction
$\mid$	$\mathbf{let}\ (a, b) = [-]\ \mathbf{in}\ e$	bind $a \cup b$ in e	pair elimination
$\mid$	$[-]e$		application
$\mid$	$v[-]$		application

Θ ::=			fractional capability environment
$\mid$	$\cdot$		
$\mid$	Θ, fc		

Γ ::=			linear types environment
$\mid$	$\cdot$		
$\mid$	$\Gamma, x : t$		
$\mid$	Γ, Γ'		

Δ ::=			intuitionistic types environment
$\mid$	$\cdot$		
$\mid$	$\Delta, x : t$		

σ ::=			heap
$\mid$	$\{\}$		empty heap
$\mid$	$\sigma \uplus \{l \mapsto_f m_{k_1,k_2}\}$		location l points to matrix m

StepsTo	::=		result of small step
	$\mid$	$\langle \sigma, e \rangle$	heap and expression
	$\mid$	**err**	error